Machine Learning Portfolio Allocation

Michael Pinelis* and David Ruppert**

*Department of Economics, Cornell University, mdp93@cornell.edu
**Department of Statistics & Data Science and School of Operations Research and Information Engineering, Cornell University, dr24@cornell.edu

March 2, 2020

Abstract

We find economically and statistically significant gains from using machine learning to dynamically allocate between the market index and the risk-free asset. We model the market price of risk to determine the optimal weights in the portfolio: reward-risk market timing. This involves forecasting the direction of next month's excess return, which gives the reward, and constructing a dynamic volatility estimator that is optimized with a machine learning model, which gives the risk. Reward-risk timing with machine learning provides substantial improvements in investor utility, alphas, Sharpe ratios, and maximum drawdowns, after accounting for transaction costs, leverage constraints, and on a new out-of-sample test set. This paper provides a unifying framework for machine learning applied to both return- and volatility-timing.

1 Introduction

We use machine learning to find the optimal portfolio weights between the market index and the risk-free asset. We model the future market price of risk as the weight of the equity index in our portfolio. The weight is proportional to the reward component, given by the return forecast based on the probability of the excess market return being positive. The market exposure is inversely proportional to the risk component, an estimate of prevailing volatility. This procedure is simultaneously return- and volatility-timing the market and can be called 'reward-risk timing'[1]. Our results document that a portfolio allocation strategy that employs machine learning to reward-risk time the market gives an 95% improvement in investor utility and earns a large alpha of 4%. We motivate our analysis from the vantage point of a utility-maximizing investor, who adjusts the allocation according to the attractiveness of the risk-reward trade-off. The results of this

[1]This term is from Kirby and Ostdiek (2012), who propose weighting by individual price of risks in a multi-asset portfolio. Our paper focuses on the portfolio with the market index and risk-free asset. Another difference is Kirby and Ostdiek (2012) use several-year-long rolling window estimates of the conditional mean and volatility while we look at the dynamic nine-months rolling data window for machine learning strategies.

paper can be applied by industry practitioners, institutional investors, or the individual investor.

A number of papers have been written on predicting returns and volatilities with machine learning and large numbers of features. See as a review (Yoo et al., 2005). Machine learning methods have been shown to be suitable and advantageous for the difficult task of identifying the regimes in the markets (Gu et al., 2018). Gu et al. find a benefit of using machine learning for market timing with return forecasts of 26% and 18% increases in Sharpe ratios relative to that of the buy-hold with neural networks and Random Forest, respectively. Our results document a 40% increase when using Random Forest for both returns and volatilities in combination. Taking advantage of the allowance for nonlinear predictor interactions in machine learning models gives better return forecasts and parameter values in a volatility estimator based on market conditions. An approach with machine learning that considers both expected return- and volatility-timing leads to a profitable trading strategy, without an extensive set of predictors. This paper studies how the machine learning method of Random Forest can forecast the sign of the risk premia with past dividend yields. Then a separate Random Forest model is employed to predict the optimal parameters of a volatility estimator. Specifically, we apply the model to estimate the volatility reference window as a function of lagged volatilities. Comparing the performance of linear regression for reward-risk timing, we show that machine learning outperforms by a significant margin.

Expected-return or reward-timing involves adjusting portfolio allocation according to beliefs about future asset returns. This is akin to benchmark timing, the active management decision to vary the managed portfolio's beta with respect to the benchmark (Grinold and Kahn, 1999). Merton (1981) derived the economic value of return forecasts. Campbell and Thompson (2008) show that many predictive regressions beat the historical average return, once weak restrictions are imposed on the signs of coefficients and return forecast.

Volatility- or risk-timing is a newer idea. While there is a wide array of volatility-based portfolio allocation strategies, this paper derives directly from the utility maximization principle a strategy that naturally depends on both the return and volatility. With this methodology, the portfolio weight in the risky asset is inversely proportional to the recent volatility, which turns out to be similar to the assumption in Moreira and Muir (2017). Intuitively, by avoiding high-volatility times the investor avoids risks, but if the risk-return trade-off is strong one also sacrifices expected returns, leaving the volatility timing strategy with no edge. Most commonly, the volatility estimator is the realized volatility for the past few months. We propose a dynamic volatility estimator that changes the look-back window length with machine learning that is based on the optimal portfolio weight. To best respond to market conditions, one needs a volatility estimator that itself responds to market conditions as well. Varying the length of the volatility reference period in the standard square-sum of squared returns formula gives a more accurate reflection of market conditions that filters out noise better than static volatility estimators. The results show that the benefits from volatility-timing are enhanced when using this proposed measure for volatility.

Reward-risk timing is the combination of both return- and volatility-timing. Return-timing can be profitable with superior forecasting ability, yet ignoring the risk associated with a high return, for instance, would lead to poor risk-adjusted performance. The incorrect forecasts are not mitigated by their risk. On the other hand, volatility-timing is advantageous if the risk is not compensated fully by the reward, yet there may be cases when in fact the reward overcompensates the risk. Timing the market with the price of risk accounts for the drawbacks of these individual approaches. The role of machine learning is to provide more accurate estimates by taking advantage of complex non-linear relationships between market variables and help make optimal decisions. With this, we provide a unifying framework for return- and volatility-timing as well as machine learning in finance.

An outline of the paper follows. Section 2 reviews the literature. Section 3 describes the portfolio allocation methodology, including the utility-maximization problem and model descriptions. Section 4 demonstrates the results of using the machine learning portfolio allocation strategy. Section 5 contains theoretical interpretations of the results, and Section 6 concludes.

2 Literature

Abundant work can be found on two strands of market timing, via expected returns and volatilites. Work can also be found on approaches combining the two, yet none to our knowledge integrate machine learning.

There is a long literature on expected-return timing. Kandel and Stambaugh (1995) examine equity return predictability and find that the optimal stock-versus-cash allocation can depend importantly on a predictor variable such as the dividend yield. Goyal and Welch (2008) comprehensively examine the performance of variables that have been suggested by the academic literature to be good predictors of the equity premium, and they find contradictory results. Johannes et al. (2014), however, find strong evidence that investors can use predictability to improve out-of-sample portfolio performance provided they incorporate time-varying volatility and estimation risk into their optimal portfolio problems.

There has also been a sizable interest in volatility-timing. Moreira and Muir (2017) showed volatility-managed factors outperform their buy-and-hold counterparts, modeling the optimal weight as a constant over the realized volatility for the previous month. Fleming et al. (2007) discussed the economic value of volatility timing, and Moreira and Muir (2019) derived that investors who volatility time earn 2.4% more annually than those who do not. Numerous papers have been written in response. Liu et al. (2019) found that Moreira and Muir's strategy is subject to look-ahead bias since they choose the constant based on the full sample and it is not easy to outperform the market with volatility timing alone. One finding in this paper is that simply replacing the constant with the expanding estimate of the unconditional mean, which stays close to the constant chosen by Moreira, leads to nearly the same performance[2]. Another criticism of Moreira and Muir's paper raised by Cederburg et al. (2019) is that a strategy that has a positive alpha will not necessarily add to the investment value for an investor; the investment value increases only if the strategy yields a greater Sharpe ratio or

[2]Our weight is constrained by a 150% leverage limit so the alphas are not the same in the main results.

higher investor utility when it is combined with the market or the investor's existing portfolio.

Our main aim is to simultaneously perform expected return- and volatility-timing. Johannes et al. (2014) find statistically and economically significant out-of-sample portfolio benefits for an investor who uses models of return predictability when forming optimal portfolios, when accounting for estimation risk and allowing for time-varying volatility. We do so, however, not with typical regression-based approaches but with machine learning.

Kirby and Ostdiek (2012) develop volatility- and reward-risk-timing strategies for the portfolio with many assets. Our paper considers the problem for the risk-free asset and the market while applying machine learning.

Gu et al. (2018) showed the benefit from using machine learning for empirical asset pricing, tracing the predictive gains to the allowance of non-linear predictor interactions. Trees and neural nets were the most successful in predicting returns.

An article by Hallac et al. (2018) proposes an approach to dynamic asset allocation using Hidden Markov Models that is based on detection of change points without fitting a model with a fixed number of regimes to the data, without estimating any parameters, and without assuming a specific distribution of the data. Our approach also does not assume a number of regimes, yet it does not discretize the portfolio weights.

To our knowledge, this is the first paper written on a machine learning approach to simultaneous return- and volatility-timing.

3 Methodology

We perform two tasks with machine learning that give the weight of the market index in our portfolio. First, we predict if the market excess return next month will be positive with lagged net payout yields and risk-free rates as the predictor variables. Second, we estimate the prevailing volatility with lagged values for a volatility proxy. The weight of the equity index is proportional to the probability that the next month's return exceeds that of the risk-free asset and inversely proportional to the volatility estimate. This gives us a series of out-of-sample portfolio returns and corresponding performance metrics. Finally, the same procedure is performed on a holdout set, data that provides a final estimate of the models' performance after they have been trained and validated, to test against backtest-overfitting (Bailey et al., 2015) [3]. Algorithm 1 describes the general portfolio allocation approach.

[3] Holdout sets are never used to make decisions about which algorithms to use or for improving or tuning algorithms. Therefore, the performance on the holdout set is indicative of investment performance if an investor starts trading with the models and strategy today.

Algorithm 1: Portfolio Allocation Approach

for *each month* $t = 1$ *to* T **do**

1. Update machine learning models with the data until the most recent returns and predictors at time $t-1$
2. Forecast the class probabilities of the sign of the excess return at time t and the optimal reference window length for the volatility estimate
3. Compute the optimal weight in the stock index for time t and return to step 1

end

The strategies begin on January, 1952. The reason for this is two-fold. First, it is important that the data that trains a machine learning model is large enough. Trying to forecast with a model that has seen too few observations is problematic. As such, one can expect the overall accuracy of the models to improve with time. Second, the unconditional mean is highly sensitive to large changes in returns when over few observations. In particular, the Great Depression period contains both extreme negative and positive returns that lead to volatile estimates.

We conduct an extensive array of tests to evaluate the robustness of our results. A key result is that the typical investor can benefit from reward-risk timing even if subject to realistic transaction costs and tight leverage constraints. A comparison of the Sharpe ratio of similar strategies that do not employ machine learning finds less impressive performance. Furthermore, examining the results of a series of time-series regressions gives additional evidence for positive alphas. Finally, we derive the theoretical alpha generation process to help explain these findings.

3.1 Portfolio Allocation

Most models of portfolio allocation with exact, closed-form solutions assume expected returns or stochastic volatility evolve continuously through time, a constant investment opportunity set, or single-period optimization. Our problem is harder due to the presence of time-varying risk premia and volatility across a discretized time horizon with periodic rebalancing. To find tractable solutions that are applicable to real-life investors, one can first consider the one-period problem in Merton (1969) and Samuelson (1969) with a constant market price of risk, the ratio of reward above the risk-free relative to risk, followed by stylized cases with time-varying volatility and time-varying price of risk which give our optimal weights.

We focus our attention on a power utility investor of terminal wealth $W_{t+\Delta t}$.

$$U(W_{t+\Delta t}) = \frac{W_{t+\Delta t}^{(1-\gamma)} - 1}{1-\gamma}, \tag{1}$$

where $\gamma > 0$ is the coefficient of relative risk-aversion. For $\gamma = 1$, $U(W_{t+\Delta t}) = \ln W_{t+\Delta t}$.

The investment universe with a risky and riskless asset and a constant market price of risk (mean and variance) constrained by a budget is defined by

$$r_t = \mu + \sigma \cdot z_t \tag{2}$$

$$W_t = W_{t-1}(w_t \cdot \exp(r_t) + (1 - w_t) \cdot \exp(r_t^f)), \tag{3}$$

where μ is the expected return on the risky asset, σ is the volatility, z_t is a normal random variable with mean zero and $E[z_t|z_{t-1}] = E[z_t]$, W_t is the investor's wealth at time t, r_t^f is the risk-free asset log return, and w_t is the portfolio weight in the risky asset at time t. While the return on the risk-free asset is realized at time t, the rate is locked in at time $t-1$. Samuelson (1969) showed the optimal investment fraction in the risky asset to maximize the expected utility of wealth is given by:

$$w_t^* = \frac{\mu - r_t^f}{\gamma\sigma^2}, \tag{4}$$

It is well known that the investment opportunities are not constant throughout time. Therefore, consider the following model where the market price of risk changes according to two non-linear functions of lagged predictor variables and volatilities.

$$r_t = \mu_t + \sigma_t \cdot z_t \tag{5}$$

$$\mu_t = g_t(x_{t-1}, ..., x_{t-9}, r_{t-1}^f, .., r_{t-5}^f) + \epsilon_t \tag{6}$$

$$\sigma_t^2 = h_t(\sigma_{t-1}^2, ..., \sigma_{t-9}^2) + s_t, \tag{7}$$

where $x_{t-1}, ..., x_{t-9}$ are the nine lagged values of predictor variable, $\sigma_{t-1}^2, ..., \sigma_{t-9}^2$ are the nine lagged volatilities, z_t, ϵ_t, and s_t are potentially correlated normal random variables with mean zero, $E[z_t|z_{t-1}] = E[z_t]$, $E[\epsilon_t|\epsilon_{t-1}] = E[\epsilon_t]$, and $E[s_t|s_{t-1}] = E[s_t]$. In certain stylized cases, there exist closed-form solutions to multi-period investment problems when variables at the current time are unknown. As Johannes et al. (2004) point out, however, for an analytical solution, expected returns can be unknown only if the current volatility is known, for instance, by the quadratic variation process. Because both future returns and volatility are predicted, to solve the optimal portfolio problem, we follow the existing literature and simplify the allocation problem by considering a single-period problem:

$$J(\mathcal{F}_{t-1}) = \max_{w_t} E[U(W_t)|\mathcal{F}_{t-1}] = \max_{w_t} \int U(W_t)P(r_t|\mathcal{F}_{t-1})dr_t, \tag{8}$$

where $P(r_t|\mathcal{F}_{t-1})$ is the predictive distribution of future returns and $\mathcal{F}_{t-1} = \{x_{t-1}, ..., x_{t-9}, r_{t-1}^f, .., r_{t-5}^f, \sigma_{t-1}^2, ..., \sigma_{t-9}^2\}$. This is similar to the approach taken in Kandel and Stambaugh (1996) and Johannes et al. (2004).

The difference between single and multi-period problems is that in the latter, hedging demands arising from changes in variables determining the attractiveness of future investment opportunities. Brandt (1999) showed that hedging demands are typically very small terms in the optimal weight. Additionally, portfolio choice will be myopic if the investor has power utility and returns are IID.

To derive the optimal portfolio weight, let us assume that $U(W_t)$ is twice differentiable, monotonically increasing, and concave in the weight (which is the case for the power utility investor). Then by Eq. 3, the optimal portfolio is given by the first order condition

$$E[U^{'}(W_t)(r_t - r_t^f)|\mathcal{F}_{t-1}] = 0, \tag{9}$$

where the expectation is taken over the predictive distribution of future returns. By the definition of covariance and Eq. 9,

$$cov[U^{'}(W_t), r_t - r_t^f|\mathcal{F}_{t-1}] + E[U'(W_t)|\mathcal{F}_{t-1}]E[r_t - r_t^f|\mathcal{F}_{t-1}] = 0, \tag{10}$$

To separate the effects of risk and return on utility, realize that W_t and $r_t - r_t^f$ are jointly normally distributed. In this case, Stein's lemma allows us to re-write the covariance term as

$$\begin{aligned} cov[U^{'}(W_t), r_t - r_t^f|\mathcal{F}_{t-1}] &= E[U^{''}(W_t)|\mathcal{F}_{t-1}]cov[W_t, r_t|\mathcal{F}_{t-1}] \\ &= w_t E[U^{''}(W_t)|\mathcal{F}_{t-1}]var[r_t|\mathcal{F}_{t-1}]. \end{aligned} \tag{11}$$

Solving for the optimal weight,

$$w_t^* = \frac{E[r_t - r_t^f|\mathcal{F}_{t-1}]}{\gamma \cdot var[r_t|\mathcal{F}_{t-1}]}, \tag{12}$$

where $\gamma = -E[U^{'}(W_t)|\mathcal{F}_{t-1}]/E[U^{''}(W_t)|\mathcal{F}_{t-1}]$. This provides a justification for using a conditional mean-variance rule.

As a final case, consider constant returns and time-varying volatility:

$$r_t = \mu + \sigma_t \cdot z_t \tag{13}$$

$$\sigma_t^2 = h_t(\sigma_{t-1}^2, ..., \sigma_{t-9}^2) + s_t \tag{14}$$

Starting from Eq. 10, using the fact that $E[r_t - r_t^f|\mathcal{F}_{t-1}] = E[r_t - r_t^f]$, and applying the same logic, the optimal weight is given by

$$w_t^* = \frac{E[r_t - r_t^f]}{\gamma \cdot var[r_t|\mathcal{F}_{t-1}]}. \tag{15}$$

The two functions $g_t(\mathcal{F}_{t-1}) = r_t - r_t^f$ and $h_t(\mathcal{F}_{t-1}) = \sigma_t^2$ give the return and variance, respectively, at time t given the information set $\mathcal{F}_{t-1}$ at the previous time. In this paper, we learn g_t and h_t with the machine learning algorithm Random Forest discussed in Section 3.3.

With this portfolio allocation framework in mind, we examine three reward-risk strategies based on the optimal weight. The first strategy is reward-risk timing with an expanding window estimate of the reward, the numerator in Eq. 15. It assumes time-varying volatility, but the investor has no superior knowledge about the risk premium, defined as the excess market return, at time t. Specifically, in this base strategy, volatility is computed as the realized volatility for the past month but the risk premia with the sample until time $t-1$. Therefore, the strategy is driven by $\frac{1}{t-1}\sum_{i=1}^{t-1}(r_t - r_t^f)/(\gamma \cdot \sigma_{t-1}^2)$, an estimate of the prevailing price of risk. The second and third, full reward-risk timing strategies employ machine learning (linear regression) models to 1) forecast the probabilities of the signs of the excess return for the next month with lagged dividend yields (the absolute return), giving the respective estimates for the numerator in Eq. 12 and 2) estimate the best length of the reference window to use in the volatility calculation with lagged volatilities, giving the variance in the denominator of Eq. 12.

Given the excess return class probabilities, the numerator in the optimal weight is adjusted with a more accurate view of the expected reward. A correct return direction prediction 55% of the time, on average, signifies an advantage over using the unconditional mean. By varying the length of the reference window, the volatility estimate can be adjusted and improved[4]. The length determines which months' realized returns are included in the estimate, and in effect the magnitude of the volatility estimate. Correctly scaling the volatility in terms of the actual future excess return is the result.

The reward-risk timing strategies avoid investing during periods of low market reward and high risk. It is not surprising that the performance of the base reward-risk timing strategy is better relative to the buy-and-hold given that it is an extension of the risk-managed portfolio literature discussed in the next subsection. The full strategy employing machine learning achieves better results than both strategies. Next, we look more closely at the volatility-timing strategy in the literature and the modification that is made to arrive at the base strategy.

3.1.1 Volatility-Timing

Moreira and Muir (2017) examine a volatility-managed portfolio constructed by scaling the portfolio weight of the market or factor w_t by the inverse of the past month's realized daily return variance. The strategy is motivated by the observation that changes in volatility over time are not offset by proportional changes in returns. The authors find that this volatility-timing strategy improves investment performance relative to the original market and factors by reducing risk exposure when volatility is high (Liu et al., 2019). In this volatility-managed portfolio, the weight in the index is inversely proportional to the volatility.

$$w_t = \frac{c}{\hat{\sigma}_{t-1}^2}, \tag{16}$$

where c is a constant and $\hat{\sigma}_{t-1}^2$ is the realized return variance in month $t-1$ computed from the 22 average daily returns over the month, $\hat{\sigma}_{t-1}^2$ is computed as

$$\hat{\sigma}_t^2(f) = RV_t^2(f) = \sum_{d=1/22}^{1}\left(f_{t+d} - \frac{\sum_{d=1/22}^{1} f_{t+d}}{22}\right)^2, \tag{17}$$

where $f = r - r^f$ is the daily excess return. The constant c is set in Moreira and Muir (2017) such that the strategy's standard deviation matches that of the buy-and hold

[4] We improve the volatility estimate in the sense that the estimate gives a higher expected return when using it to determine portfolio weights.

for ease of interpretation. Liu et al. (2019) point out that choosing c based on the unconditional volatility over the entire period is an in-sample approach and is thus subject to look-ahead bias. While this is correct, simply using the historical average excess return forecast as the constant gives the same or better performance results. This is not surprising since the historical mean divided by the risk-aversion coefficient $\gamma = 4$ produces a numerator that stays consistently close to the the exact value of c, the constant which makes the standard deviation of the volatility-managed strategy equal to that of the buy-and-hold[5]. Figure 1 displays the unconditional mean scaled by $1/\gamma$ from 1952 to 2010, along with the value of c.

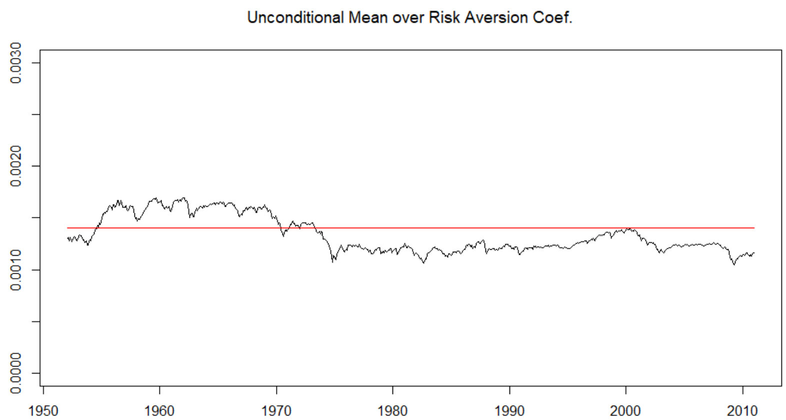

Figure 1: Volatility-timing with a constant versus with expanding window estimate of excess return. The constant c, which gives the volatility-timing strategy the same ending standard deviation as the buy-and-hold, is plotted in red versus the numerator obtained from using the expanding excess return mean over a risk-aversion coefficient $\gamma = 4$ in black.

The historical estimate slightly exceeds c for most of 1952 to 1970 and falls below c for the rest of the period.

The discussion above provides an intuition for why this modified version of volatility-timing, or base reward-risk timing, achieves similar investment performance for the market portfolio to volatility-timing in Moreira and Muir (2017). The results are discussed in Section 4. To come to the full strategy, we first look at the regression and machine learning models in the next sections.

3.2 Regression

We consider a linear regression model with extensions as a comparison to machine learning. Only the first task of risk premia estimation is used in this comparison, with the volatility estimate of this strategy set equal to that of the machine learning model.

Starting with the simple model of monthly excess returns, f_t, as a function of the

[5]Because our data has a slightly shorter sample period, the value here does not exactly match that in the papers above.

lagged payout yields and risk-free rates,

$$f_t = \alpha + \sum_{i=1}^{9} \beta_i x_{t-i} + \sum_{j=10}^{14} \beta_j r^f_{t-j+9} + \epsilon_t \tag{18}$$

we find that the residuals are serially correlated. For this reason, we model the residuals as an ARMA process.

$$\epsilon_t = \phi_1 \epsilon_{t-1} + \cdots + \phi_p \epsilon_{t-p} + \theta_1 z_{t-1} + \cdots + \theta_q z_{t-q} + z_t, \tag{19}$$

where z_t is white noise. The number of AR and MA terms, p and q, are chosen at each time with AICc. One alternative to this specification is an ARMAX model that is estimated with maximum likelihood. However, the coefficients are harder to interpret. Regression with ARMA errors can capture the residual persistence, if it is present, while allowing rapid changes in the dependent variable. This modification slightly improves predictive performance.

3.3 Random Forest

A Random Forest is an ensemble machine learning algorithm developed by Breiman (2001). The prediction by the Random Forest is the majority vote across all the individual decision tree learners (Hastie et al., 2017). The default tree bagging procedure draws B different bootstrap samples of the training data and fits a separate classification tree to the bth sample. The forecast is the average of the trees' individual forecasts. Trees for a bootstrap sample are usually deep overfit, meaning each has low bias but is inefficiently variable. Averaging over B predictions reduces the variance and stabilizes the trees' forecast performance. Algorithm 2 gives the procedure used to construct a Random Forest with the implementation by Liaw and Wiener (2002).

Algorithm 2: Random Forest

Result: The ensemble of trees $\{T_b\}^B$
for $b = 1$ *to* B **do**

1. Draw a bootstrap sample $\mathbf{Z}^*$ of size n from the training data.
2. Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size s_{min} is reached.
    (a) Select m variables at random from the p variables
    (b) Pick the best variable/split-point among the m.
    (c) Split the node into two child nodes.

To make a prediction at a new point, $\vec{x}$, let $\hat{C}_b(\vec{x}) \in \{-1, 1\}$ be the class prediction of the bth random-forest tree, and then $\hat{C}(\vec{x}) = sign(\sum_{b=1}^{B} \hat{C}_b(\vec{x}))$, the weighted majority vote. For a binary model, the class probabilities are $\sum_{i=1}^{B+} \hat{C}_i(\vec{x}) / \sum_{b=1}^{B} \hat{C}_b(\vec{x})$ and $\sum_{j=1}^{B-} \hat{C}_i(\vec{x}) / \sum_{b=1}^{B} \hat{C}_b(\vec{x})$, the proportion of votes for each class, with $B^+ + B^- = B$.

Random forests give an improvement over bagging with a variation designed to reduce the correlation among trees grown from different bootstrap samples. If most of the bootstrap samples are similar, the trees trained on these sample sets will be highly correlated. Then the average estimators of similar decision trees can be more robust but do not perform much better than a single decision tree. If, for example, last month's dividend yield is the dominant predictor of the return direction out of the variables, then most of the bagged trees will have low-depth splits on the most recent yield, resulting in a large correlation among their predictions. Trees are de-correlated with a method known as "random subspace" or "attribute bagging," which considers only a random subset of m predictors out of p for splitting at each potential branch. In the example, attribute bagging will ensure early branches for some trees will split on predictors other than the most recent dividend yield. Since each tree is grown with different sets of predictors, the average correlation among trees further decreases and the variance reduction relative to standard bagging is larger (Gu et al. 2018)[6]. The number of variables allowed to choose from m, number of bootstrap samples B, and the minimum fraction of observations in the terminal nodes s_{min} are the tuning parameters optimized with validation. A detailed algorithm for classification trees can be found in the Appendix.

The parameters m and s_{min} are tuned with the sample from 1952 to 2010. To test against parameter over-fitting, the final values are kept on the holdout time period from 2011 to 2017.

3.4 Risk Premia Direction Prediction

Forecasting absolute returns is explored extensively in Gu et al. (2018). For better model intelligibility, we consider an alternative task, predicting whether excess returns will be positive or negative, which is a binary classification problem. For optimal portfolio construction, the weight should increase when the investor expects a positive excess return and vice versa, holding all else constant.

To classify each month, we borrow from the standard literature and use the lagged dividend payout ratios as the predictors. The importance of the dividend yield in the allocation is robust to the "data-mining" consideration (Kandel and Stambaugh, 1995), and it has been shown to explain equity return predictability in Johannes et al. (2004) for example. In traditional theory, the dividend yield can explain equity prices since prices are the discounted future cash flows.

The probability that the sign of the excess return, Y_t, will be positive or negative is

$$P(Y_t = y_t) = g_t(x_{t-1}, ..., x_{t-9}, r^f_{t-1}, ..., r^f_{t-5}) = \frac{\sum_{j=1}^{By_t} \hat{C}_j(\vec{x})}{\sum_{b=1}^{B} \hat{C}_b(\vec{x})}, \tag{20}$$

where y_t is $+$ or $-$, g_t is the Random Forest model fit at time t, $x_{t-1}, ..., x_{t-9}$ are the nine last values of the payout yield, $\hat{C}_b$ is an individual decision tree, and $\vec{x}$ is the feature vector consisting of the payout yields and risk-free rates[7]. With our portfolio

[6]Because this makes Random Forest a non-deterministic algorithm, we average the results for multiple different seeds.

[7]While the class vote proportions are not exactly the the class probabilities, we use them as a proxy.

allocation strategy, being correct more than half of the time is sufficient for the investor to benefit. There is information in the dividend yield up to three quarters ago and the presence of interaction effects between dividend yields at different months. In traditional literature, a higher past month's dividend yield is indicative of a higher chance of a positive excess return (Fama and French, 1988). Yet the yield in the month before that still has information about the overall trend in the market. We trace the predictive gains of our approach to these reasons.

For the base reward-risk timing strategy, the expected excess return $E[r - r^f]$ in w^* is kept as the mean of the expanding window of excess returns until time $t - 1$, $\overline{r - r^f} = \frac{1}{t-1}\sum_{i=1}^{t-1}(r_i - r_i^f)$. If an investor knows with some probability P and some level of confidence $1 - \delta$ that the excess return will be positive or negative, the investor can adjust the expectation to

$$\begin{aligned} E[r_t - r_t^f | P(r_t - r_t^f > 0), \delta_t] = \\ (1 - \delta_t) \cdot (\overline{r - r^f}^{+} \cdot P(r_t - r_t^f > 0) \cdot \pi_+ + \overline{r - r^f}^{-} \cdot P(r_t - r_t^f \leq 0) \cdot \pi_-) \\ + \delta_t \cdot \overline{r - r^f}, \end{aligned} \tag{21}$$

where π is the proportion of returns that were historically positive or negative multiplied by two, $\overline{r - r^f}^{+}$ and $\overline{r - r^f}^{-}$ are the means conditional on a positive or negative excess return, and $P(r_t - r_t^f > 0) + P(r_t - r_t^f \leq 0) = 1$. Here, δ_t is the test accuracy rate of the Random Forest model[8]. The fitted models are able to predict the correct direction of the return approximately 55% of the time. In other words, the numerator of the weight becomes the sum of the conditional expectations weighted by class prediction probabilities and the expectation without any knowledge of the future. The sum of conditional expectations and the unconditional expectation is itself weighted by the confidence in the machine learning model. The numerator is equal to the unconditional expectation when the probabilities of positive and negative excess returns are equal. Using a weighted average of the historical mean and Random Forest prediction reduces the frequency of large shifts in the portfolio yet allows for the share in the equity index to grow when the model is highly confident that the equity premium will be high. For the full reward-risk timing portfolio, the expectation of the risk premia is Eq. 21.

3.5 Volatility Estimation

Volatility has a central role in optimal portfolio selection, derivatives pricing, and risk management. These applications motivate an extensive literature on volatility modeling. Starting with Engle (1982), researchers have fit a variety of autoregressive conditional heteroskedasticity (ARCH), generalized ARCH (Bollerslev, 1986), and stochastic volatility models to asset returns (Fleming et al., 2001). GARCH models are widely used for their ability to permit a wide range of behavior, in particular, more persistent periods of high or low volatility than seen in an ARCH process (Ruppert and Matteson, 2015). We choose an alternate route and model the discrete parameter of

[8]The accuracy rate fluctuates slightly at each iteration and is therefore updated with the expanding window of predictions until the current time.

the simple volatility estimator defined as the standard deviation of the past N daily log returns. The motivation behind this approach is the varying choice of N in the risk-managed factors literature[9].

$$\sigma_t = \sqrt{\frac{22}{N}\sum_{d=1}^{N}\left(f_{t+1-d/N} - \frac{\sum_{d=1}^{N} f_{t+1-d/N}}{N}\right)^2}. \tag{22}$$

The number of returns to use in the volatility calculation, N, is the output of the Random Forest model trained on an expanding window of data until time $t-1$ and is restricted to values that include no partial month so the problem is multi-class classification. Another restriction on the values of N is 1 month or multiples of 3 months until 9, i.e. $\frac{N}{22} \in \{1, 3, 6, 9\}$ to limit the frequency of changes. Since the optimal weight of the market index at time t is inversely proportional to the volatility, the optimal number of returns to include in the estimate, N_t^*, is defined as the value which makes the volatility estimate the maximum or the minimum under the previous constraints depending on the sign of the excess return:

$$\text{If } r_t > r_t^f, N_t^* := \arg\min_N \tfrac{22}{N}\textstyle\sum_{d=1}^{N}\left(f_{t+1-d/N} - \frac{\sum_{d=1}^{N} f_{t+1-d/N}}{N}\right)^2,$$
$$\text{else } N_t^* := \arg\max_N \tfrac{22}{N}\textstyle\sum_{d=1}^{N}\left(f_{t+1-d/N} - \frac{\sum_{d=1}^{N} f_{t+1-d/N}}{N}\right)^2.$$

σ_t with N_t^* is the return-maximizing volatility σ_t^* in our portfolio allocation framework. The future excess return, and thus N_t^*, is unknown at time $t-1$. We can, however, estimate the relationship between past values of N_t^* and some predictor variables at time $t-1$.

The predictor variables are lagged realized volatilities for the past nine months acting as proxies, $\hat{\sigma}_t = \sum_{d=1}^{22}\left(f_{t+1-d/22} - \frac{\sum_{d=1}^{22} f_{t+1-d/22}}{22}\right)^2$, and

$$N_t = h_t(\hat{\sigma}_{t-1}^2, ..., \hat{\sigma}_{t-9}^2). \tag{23}$$

The reference window length is a function of the lagged volatilities. Thus, the investor's estimate of the squared return-maximizing volatility σ_t^{*2} given the estimated optimal reference window length $\hat{N}_t$ becomes

$$E[\sigma_t^{*2}|\hat{N}_t] = \bar{\sigma}_{t-1}^2, \tag{24}$$

where $\bar{\sigma}_{t-1}^2 = \frac{22}{\hat{N}_t}\sum_{d=1}^{\hat{N}_t}\left(f_{t-d/\hat{N}_t} - \frac{\sum_{d=1}^{\hat{N}_t} f_{t-d/\hat{N}_t}}{\hat{N}_t}\right)^2$. If $\hat{N}_t = 22$, the average number of trading days in a month, the volatility estimate is simply equal to the last month's

[9]Barroso and Santa-Clara (2014) use a 6-month estimate of realized volatility to construct their risk-managed momentum strategy. On the other hand, Moreira and Muir (2017) use a single month for a number of factors including momentum, indicating the choice for N could be optimized. We delegate the decision to machine learning and find an advantage to automating the task based on prevailing market conditions.

realized volatility. The advantage of this measure over simply the last month's volatility is that it contains information about the future excess return. The majority of the time, N_t takes either the values of 1 month or 9 months, and changes in the window length are usually persistent.

The test, or out-of-sample, accuracy for the Random Forest is defined as

$$\Delta = \frac{1}{t}\sum_{i=1}^{t} \mathbb{1}\{h_i(\cdot) = N_i^*\}. \tag{25}$$

The accuracy of these Random Forest models is on average 40% because classification correctness is a harsh metric for multi-class models. Because there are four classes, the 40% attained by the model should be measured against 25% on average with random guessing and is a substantial improvement. This accuracy, however, is sufficient for a benefit in performance. See Section 4.

4 Empirical Results

4.1 Data Description

This paper uses monthly data from Kenneth French's website on the market return (Mkt) and risk-free asset return (Rf). Daily returns are retrieved to compute the realized volatilties.

We also use data on the payout yield from Michael Robert's website, which is derived from all firms continuously listed on the NYSE, AMEX, or NASDAQ indices. The payout yield here is a more inclusive measure of total payouts than standard dividend yields and is achieved via the 'net payout' of Boudoukh et al. (2007). It includes share issuances and repurchases in addition to the traditional cash dividend yields. For the payout yield after 2010, CRSP monthly data at the firm-level and the same aggregation procedure to form the yields is used.

4.2 Predictive Performance

To assess the predictive performance for excess market return forecasts, we calculate the out-of-sample R^2 over 1952-2017 as

$$R^2_{oos} = 1 - \frac{\sum_{t\in\mathcal{T}}(f_t^A - \hat{f}_t^A)^2}{\sum_{t\in\mathcal{T}} f_t^{A2}} \tag{26}$$

where $\mathcal{T}$ denotes the set of points not used for model training and f^A are annual market excess returns. The forecasts, $\hat{f}^A$, are formed with an average of the monthly forecasts.

The annual R^2_{oos} is 19.0% for the Random Forest model. This is 14.3% greater than that of linear regression. Gu et al. (2018) attain an annual, stock-level out-of-sample R^2 of 15.7% with Neural Networks on an optimized set of predictors.

In papers with market forecasting applications, the mean squared forecast error (MSFE) is often used to measure statistical accuracy. We explain why this is not an appropriate measure for our models.

First, our objective is to maximize investor utility and risk-adjusted returns, not predict the precise magnitude of returns. To attain a good monthly MSFE, forecasts must match the magnitude approximately. Our forecasts, given by a sum of the historical negative and positive returns weighted by probabilities, are less variable and stay close to the long-run mean. Subtracting these small fitted values from the actual returns and summing the squared results gives a MSFE that is essentially insensitive to model quality. Second, the expected return forecast is only part of the optimal trading strategy. The excess return only composes the numerator in the optimal weight given by Eq. 12. With these forecasting characteristics in mind, we next discuss the risk-adjusted performance of the strategies and models.

4.3 Risk-Adjusted Returns

This section discuss the out-of-sample investment performance for machine learning reward-risk timing and makes the relevant comparisons. We invest $1 in 1952 as an investor with a coefficient of relative-risk aversion $\gamma = 4$ and plot the cumulative returns to each strategy on a log scale in Figures 2 and 3 without short-selling and with 100% and 50% leverage constraints, respectively. For the rest of the paper, we impose the more realistic portfolio constraint, preventing the investor from taking more than 50% leverage as in Campbell and Thompson (2008): that is, confining the portfolio weight on the market index to lie between 0% and 150%.

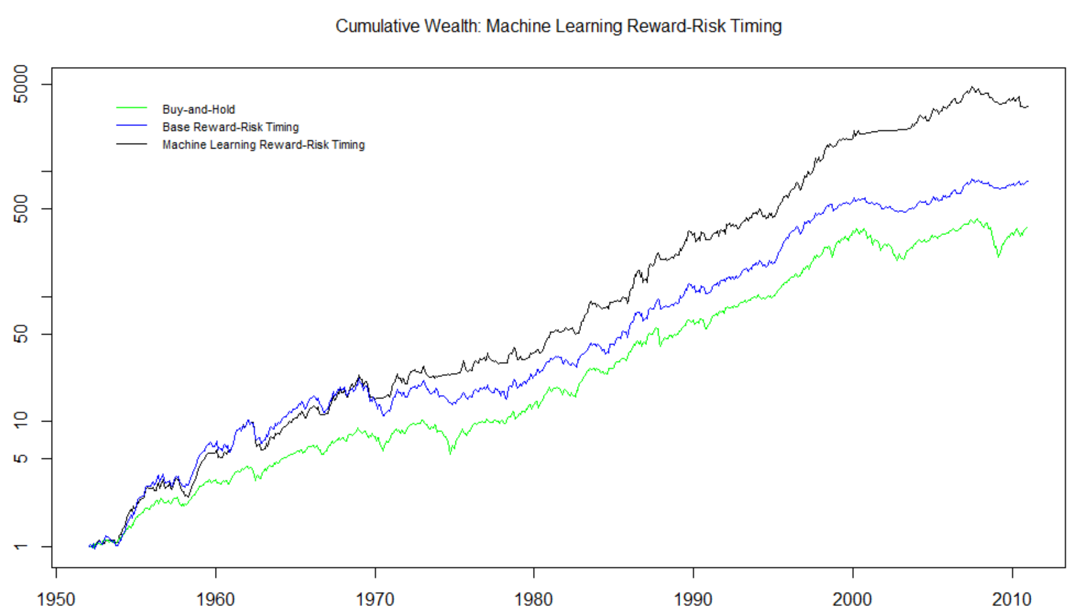

Figure 2: Cumulative returns of reward-risk timing to market index (200% leverage limit). This figure plots the cumulative returns of the base reward-risk timing strategy in blue and machine learning reward-risk timing in black against the market index in green from 1952 to 2010. The vertical axis is in log-scale.

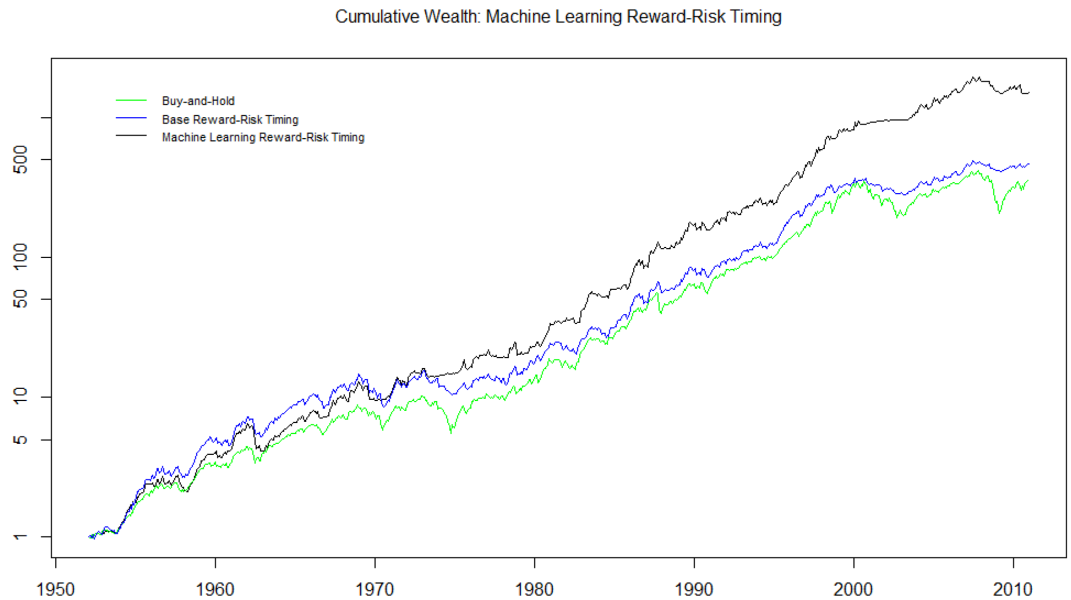

Figure 3: Cumulative returns of reward-risk timing to market index (150% leverage limit). This figure plots the cumulative returns of the base reward-risk timing strategy in blue and machine learning reward-risk timing in black against the market index in green from 1952 to 2010. The vertical axis is in log-scale.

The investments that reward-risk time realize relatively steady gains. The final wealth accumulates to around $1,500 and $500 at the end of the sample for the machine learning and base (expanding sample mean reward estimate and previous month realized volatility risk estimate) strategies, respectively, versus about $400 for the buy-and-hold. At the start of the period, the machine learning models have seen three-hundred observations as part of the training data, and the investment performance improves with the size of the training set and the classification accuracy becomes more stable. The 'break-away' moment from the base reward-risk timing strategy is around 1970. Because the Random Forest models' parameters are determined within this period, it is necessary to also look at the cumulative returns for the holdout period from 2011 to 2017 in Figure 4.

An investors who starts with $1 in 2011 and reward-risk times with machine learning achieves outperformance relative to the market and other strategies again. Therefore, the results cannot be easily explained by the particular choice of machine learning model parameters.

Figure 5 plots the drawdown of the two strategies relative to the market, which helps us understand when our strategies lose money relative to the buy-and-hold. The base reward-risk strategy take relatively more risk when volatility is low (e.g., the 1970s) and thus, not surprisingly, its largest losses are concentrated in these times. The machine learning analog has a pattern of losses close to reward-risk timing with no

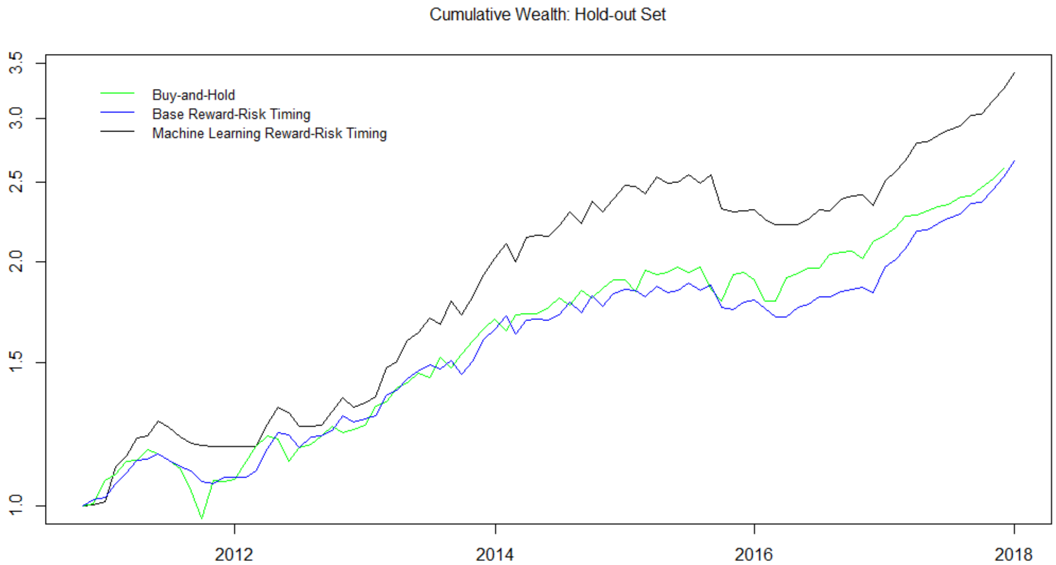

Figure 4: Cumulative returns of reward-risk timing to market index (150% leverage). This figure plots the cumulative returns of the base reward-risk timing strategy in blue and machine learning reward-risk timing in black against the market index in green from 2011 to 2017, the unseen sample period. The vertical axis is in log-scale.

predictive model, yet it diminishes the severity of many losses and to a high degree for some of the most extreme negative returns. For the sharp market losses starting in 1962, the first major drawdown, the return direction machine learning model's response is delayed, due to the very sudden drop. Yet for the next two major drawdowns in 1969 and 1973, our machine learning models are able to recognize the incoming negative returns because the drops are more staggered, cutting the losses felt by investors greatly. This is seen even more clearly in the Dot-com bubble, where using machine learning allows investors to almost completely avoid losses during this time. In the last recession of 2007-2008, due to the extremely sharp onset, our return direction machine learning models reduce risk exposure slightly too late, yet the information in the volatility estimate still correctly steers market exposure down. Reward-risk timing never has a drawdown greater than 40% of the portfolio value and greatly mitigates three of the four largest losses during severe recessions.

Before proceeding with the numerical results, we define the various strategy weights and give descriptions:

- $w_1 = max(min(E_{RF}[r - r^f|\mathcal{F}_{t-1}]/(\gamma \cdot E_{RF}[\sigma^{*2}|\mathcal{F}_{t-1}]), 1.5), 0)$. This is using Random Forest for the reward and risk estimates and a leverage limit of 50%.

- $w_2 = max(min(E_{LR}[r-r^f|\mathcal{F}_{t-1}]/(\gamma \cdot E_{RF}[\sigma^{*2}|\mathcal{F}_{t-1}]), 1.5), 0)$. This is using regression for the reward estimate and Random Forest for the risk estimate and a leverage limit of 50%.

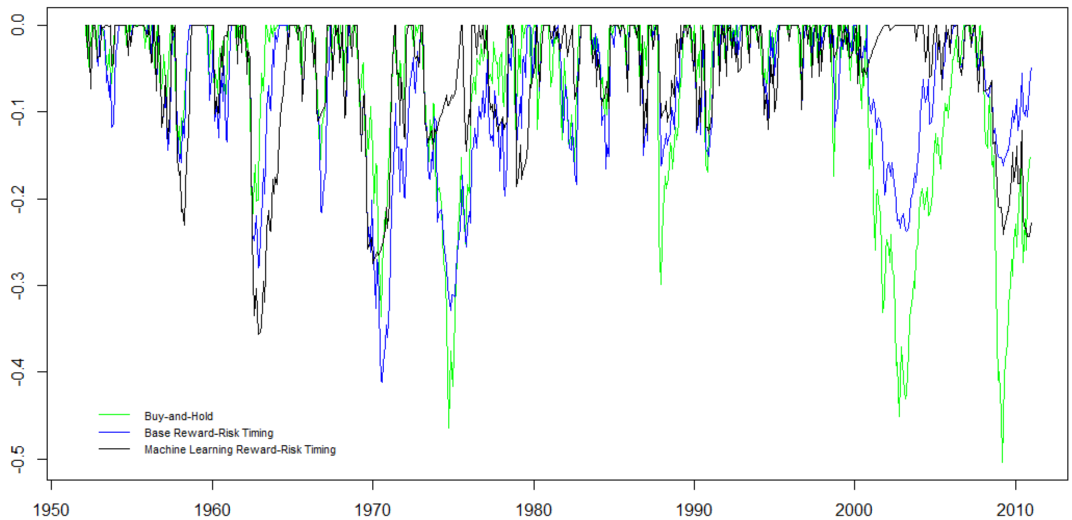

Figure 5: Drawdowns of reward-risk timing to market index. This figure plots the drawdown of the base reward-risk timing strategy in blue, machine learning reward-risk timing in black against the market index in green from 1952 to 2010.

- $w_3 = max(min(E[r - r^f]/(\gamma\sigma^2_{t-1}), 1.5), 0)$. This is using the expanding window estimate as the reward and the previous month's realized volatility as the risk (discussed in Section 3.1), with a leverage limit of 50%.
- w_4 is the same as w_1 but the 1.5 limit is decreased to 1, no leverage.
- w_5 is w_2 after the same change as in w_4.
- w_6 is w_3 after the same change as in w_4.

The risk-adjusted returns from machine learning portfolio allocation are substantially higher than reward-risk timing with no model and the buy-and-hold. Table 1 displays the Sharpe ratios for each portfolio allocation strategy and different time periods. The sample from 2011 to 2017 is a holdout set, meaning we run the portfolio allocation process on it with the same parameters and seeds as the previous sample after they are finalized.

All the active strategies outperform the buy-and-hold on a risk-adjusted basis for each out-of-sample period. Reward-risk timing with Random Forest gives the highest Sharpe ratio of 0.60 from 1952-2010, which is a 40% increase from the buy-and-hold. An investor who reward-risk times with machine learning gains more than 2 percentage points on return per year relative to passively investing, without increasing the risk.

To quantify the economic relevance of our results and facilitate comparison, we consider the perspective of a mean-variance investor. From the frame of an investor

Table 1: Sharpe Ratios

In this table are the out-of-sample annual returns, standard deviations, and Sharpe ratios for the initial period from 1952 to 2010, the holdout period from 2011 to 2017, and the full sample period for the various strategies. Mkt denotes the buy-and-hold.

Sample	Strategy	Annual Return (%)	Standard Deviation (%)	Sharpe Ratio
	Mkt	11.17	15.05	0.43
1952 -	w_3	11.53	14.82	0.47
2010	w_2	11.08	12.24	0.53
	w_1	13.55	14.92	0.60
	Mkt	14.22	11.29	1.25
2011 -	w_3	14.02	9.21	1.51
2017	w_2	13.39	8.75	1.51
	w_1	18.05	11.61	1.54
	Mkt	11.52	14.70	0.50
1952 -	w_3	11.84	14.33	0.53
2017	w_2	11.32	11.72	0.61
	w_1	14.15	14.72	0.68

with mean-variance utility, the percentage utility gain of a strategy with Sharpe ratio a from one with Sharpe ratio b is

$$\Delta U_{MV}(\%) = \frac{SR_a^2 - SR_b^2}{SR_b^2}. \tag{27}$$

From Eq. 27, a mean-variance investor who trades the market index with the help of machine learning and reward-risk timing increases lifetime utility by 95% relative to buying and holding the index.

Machine learning reward-risk timing generates large gains relative to solely focusing on either the reward or the risk component. Campbell and Thompson (2008) estimate that the utility gain of timing expected returns is 35% of lifetime utility. Moreira and Muir (2017) find that a mean-variance investor who can only trade the market portfolio can increase lifetime utility by 65% through volatility timing. Next, we run a series of time-series regression of the strategies on each other and the market index,

$$f_{t+1}^a = \alpha + \beta f_{t+1}^b + \epsilon_{t+1}, \tag{28}$$

where f_{t+1} are the monthly excess returns. A positive intercept implies that the strategy a increases Sharpe ratios relative to strategy b. When this test is applied to systematic factors (e.g., the market portfolio) that summarize pricing information for a wide cross-section of assets and strategies, a positive alpha implies that our portfolio-allocation strategy expands the mean-variance frontier.

Table 2 reports results from running regressions of the machine learning reward-risk timing strategy on the market index and the other strategies. The intercepts

Table 2: Strategy Alphas

In this table, we run time-series regressions of each strategy on the market and on one another $f^a_{t+1} = \alpha + \beta f^b_{t+1} + \epsilon_{t+1}$. The superscripts denote the three variations of strategies: RF for random forest, LR for linear regression, and base using no model. The data are monthly and the sample period is 1952 to 2010. Standard errors are in parentheses and are adjusted for heteroskedasticity (White, 1980). The alphas and errors are annualized in percent per year by multiplying monthly values by 12.

Univariate Regressions

f_a	f_b	Beta (β)	Alpha (α)	R^2	N_{obs}
Mkt^{RF}	Mkt	0.74	4.13	0.56	706
		(0.05)	(1.32)		
Mkt^{RF}	Mkt^{Base}	0.85	2.75	0.70	706
		(0.04)	(1.08)		
Mkt^{RF}	Mkt^{LR}	0.78	3.71	0.37	706
		(0.06)	(1.56)		
Mkt^{LR}	Mkt	0.50	3.18	0.37	706
		(0.04)	(1.32)		
Mkt^{Base}	Mkt	0.83	1.47	0.72	706
		(0.04)	(1.08)		

(Jensen's α's) (Jensen, 1968) are positive and statistically significant in all cases, except for the base. The machine learning strategy has an annualized alpha of 4.13% and a beta of only 0.74. The machine learning strategy over base and linear regression reward-risk timing has annualized alphas of 2.75% and 3.71%, respectively. For the comparisons, the regressions also look at the alphas attained from using linear regression and model to forecast the excess return. The alphas earned are markedly smaller at 3.18% and 1.47%, respectively.

The next finding is that our strategies survive transaction costs, given in Table 3. Specifically, we evaluate our portfolio allocation strategy for the reward-risk timing portfolios when accounting for empirically realistic transaction costs as in (Moreira and Muir, 2017). Strategies that capture reward-risk timing but reduce trading activity include capping the strategy's leverage at 1 compared to the case with a weight limit of 1.5. These leverage limits reduce trading and hence total transaction costs. We report the average absolute change in monthly weights, expected return, and Jensen's alpha of each strategy before transaction costs. The next columns contain the alphas when including various transaction cost assumptions. Finally, the last column derives the implied trading costs in basis points such that the alphas are zero in each of the cases.

The results indicate that machine learning reward-risk timing survives transactions costs, even with high volatility episodes where such fees rise. Overall, the annualized alpha of the reward-risk timing portfolio allocation strategy decreases slightly, but is still very large. Reward-risk timing with machine learning does not require extreme leverage or drastic portfolio rebalancing to be profitable.

The empirical results overall indicate a significant advantage in using machine

Table 3: Transaction Costs of Machine Learning Portfolio Allocation

In this table, we evaluate our reward-risk timing strategies for the market when including transaction costs. Lower leverage limits reduce trading activity. Specifically, we consider restricting risk exposure to be between 0 and 1 (i.e., no leverage) or 1.5. The alphas are reported with these assumptions. Following Moreira and Muir (2017), the 1bp cost comes from Fleming et al. (2003), the 10bps is from Frazzini, Israel, and Moskowitz (2015) when trading approximately 1% of daily volume, and the next column adds an additional 4bps to cover for transaction costs increasing in high-volatility episodes. The last column backs out the implied trading costs in basis points needed to drive the alphas to zero in each of the cases.

				α After Trading Costs			
Weight	$\lvert\Delta w\rvert$	$E[R]$	α	1bps	10bps	14bps	Break Even
w_1	0.27	13.55%	4.13	3.78	3.43	3.27	118 bps
w_2	0.54	11.08%	3.18	3.12	2.53	2.27	49 bps
w_3	0.33	11.53%	1.47	1.45	1.19	1.08	57 bps
w_4	0.20	11.02%	2.75	2.73	2.51	2.41	75 bps
w_5	0.37	9.05%	1.96	1.91	1.51	1.33	30 bps
w_6	0.12	9.71%	0.67	0.66	0.52	0.47	32 bps

learning for portfolio allocation. With only standard predictor variables, reward-risk timing with machine learning models offers economically substantial improvements in risk-adjusted returns (40% increase in Sharpe ratio). Statistically significant positive alphas of 4% are found as a result of the superior forecasting ability of machine learning. Finally, realistic trading costs are applied to gain further insight on real-life applicability, showing alphas remain large. With this evidence in mind, it is also valuable to look from a theoretical perspective at why the strategy outperforms.

5 Theoretical Alpha Generation Process

In this section, we provide a theoretical framework to interpret some of our findings. We first derive the alpha for the base reward-risk timing. Then we do the same process for machine learning reward-risk timing. Intuitively, the alphas for base portfolio allocation are proportional to the covariance between the conditional variance and the asset price of risk. Our alphas for portfolio allocation with machine learning are a function of models' performance.

We work in continuous time. Consider the total portfolio value process R_t with expected return r_t and conditional volatility σ_t. Then $dR_t = r_t \cdot dt + \sigma_t \cdot dz_t$. Construct the reward-risk timing version of this return with $w_t = \frac{\overline{r-r^f}}{\gamma\sigma_t^2}$ from Eq. 15,

$$\begin{aligned} dR_t' &= dR_t \cdot w_t + r_t^f dt \cdot (1 - w_t) \\ &= (dR_t - r_t^f dt) \cdot \frac{\overline{r - r^f}}{\gamma\sigma_t^2} + r_t^f dt, \end{aligned} \tag{29}$$

where r_t^f is the instantaneous risk-free rate and $\overline{r - r^f} = \frac{1}{t}\sum_{i=1}^{t}(r_i - r_i^f)$ is the expanding sample mean. The α of a time-series regression of the market-timing portfolio excess return $dR_t^{'} - r_t^f dt$ on the market portfolio excess return $dR_t - r_t^f dt$ is given by

$$\alpha = E[dR_t^{'} - r_t^f dt]/dt - \beta E[dR_t - r_t^f dt]/dt \tag{30}$$

Using that $E[dR_t^{'} - r_t^f dt]/dt = \overline{r - r^f} \cdot E[\frac{r_t - r_t^f}{\sigma_t^2}]$, $\beta = \frac{\overline{r - r^f}}{\gamma E[\sigma_t^2]}$ by minimizing the sum of squared deviations, and $E[dR_t - r_t^f dt]/dt = E[r_t - r_t^f]$ and simplifying, we obtain a relationship between the alpha and the covariance between the volatility and the price of risk.

$$\begin{aligned} \alpha &= E[\frac{r_t - r_t^f}{\sigma_t^2}] \cdot \overline{r - r^f} - E[r_t - r_t^f] \cdot \frac{\overline{r - r^f}}{E[\sigma_t^2]} \\ &= -\frac{\overline{r - r^f}}{\gamma E[\sigma_t^2]} \cdot cov[\sigma_t^2, \frac{r_t - r_t^f}{\sigma_t^2}] \end{aligned} \tag{31}$$

Thus, the α is positive when the price of risk moves opposite to the volatility. This is essentially the same result that Moreira and Muir (2017) recover. The difference here is that the alpha is amplified by the value of the sample mean of the excess return at time t rather than a constant.

Now, we examine the machine learning reward-risk timing alpha generation process:

$$dR_t^{''} = (dR_t - r_t^f dt) \cdot \frac{E[r_t - r_t^f | \mathcal{F}_{t-1}]}{\gamma E[\sigma_t^2 | \mathcal{F}_{t-1}]} + r_t^f dt, \tag{32}$$

where $\frac{E[r_t - r_t^f | \mathcal{F}_{t-1}]}{E[\sigma_t^2 | \mathcal{F}_{t-1}]}$ is the estimate of the market price of risk the models give with the information set $\mathcal{F}_{t-1}$. The α of a time-series regression of the machine learning market-timing portfolio excess return $dR_t^{''} - r_t^f dt$ on the market portfolio excess return $dR_t - r_t^f dt$ is again

$$\alpha = E[dR_t^{''} - r_t^f dt]/dt - \beta E[dR_t - r_t^f dt]/dt \tag{33}$$

Using that $E[dR_t^{''} - r_t^f dt]/dt = E[\frac{r_t - r_t^f}{\sigma_t^2}] \cdot E[r_t - r_t^f]/\gamma$ by iterated expectations and independence of the excess return and the machine learning price of risk estimate, $\beta = \frac{E[r_t - r_t^f | \mathcal{F}_{t-1}]}{\gamma E[\sigma_t^2 | \mathcal{F}_{t-1}]}$, and $E[dR_t - r_t^f dt]/dt = E[r_t - r_t^f]$, we obtain a relationship between the alpha and the price of risk expectations.

$$\alpha = \frac{E[r_t - r_t^f]}{\gamma} \cdot (E[\frac{r_t - r_t^f}{\sigma_t^2}] - \frac{E[r_t - r_t^f | \mathcal{F}_{t-1}]}{E[\sigma_t^2 | \mathcal{F}_{t-1}]}). \tag{34}$$

In this case, α is positive when the machine learning expectation of the market price of risk given the information set at the previous time is expected to be cheaper than the unconditional expectation of the price, if the excess return is positive. If the excess return is negative, then α is positive if the machine learning estimate is less than the unconditional, avoiding a high allocation. The result does not depend on the sign of the return. Not surprisingly, the alpha is positive if the accuracy of the models used to estimate the market price of risk is good enough to distinguish between positive and negative risk premia based on the known information set.

The above results provide a mapping between machine learning reward-risk timing alphas and the dynamics of the price of risk for an individual asset.

6 Conclusion

Machine learning portfolio allocation offers large risk-adjusted returns and is feasible to implement in real-time. We perform both return- and volatility-timing, or reward-risk timing, with and without machine learning, showcasing the relative advantage machine learning can provide. Furthermore, our strategy's performance is informative about the alpha generation process for actively managed portfolios.

At the same time, there are possibilities for improvements. Other machine learning methods like deep neural networks and gradient boosted trees may allow trading some interpretability for performance gains. Using predictors beyond lagged dividend yields and risk-free rates may also be beneficial. Since one of our goals here was to show that machine learning has an advantage in finance and portfolio allocation outside the context of big data, the results with standard variables are promising.

References

[1] Bailey, David, Jonathan Borwein, Marcos Lopez de Prado and Qiji Jim Zhu, 2015, The probability of backtest overfitting, In *Journal of Computational Finance*, Forthcoming.

[2] Bao, Yong, 2009, Estimation Risk-Adjusted Sharpe Ratio and Fund Performance Ranking Under a General Return Distribution, In *Journal of Financial Econometrics* 7 No. 2, 152-173.

[3] Barroso, Pedro, Pedro Santa-Clara, 2014, Momentum has its moments, In *Journal of Financial Economics* 116, 111-120.

[4] Boudoukh, Jacob, Roni Michaely, and Matthew Richardson, 2007, On the Importance of Measuring Payout Yield: Implications for Empirical Asset Pricing, In *The Journal of Finance* 62, 877-915.

[5] Bollerslev, Tim, 1986, Generalized Autoregressive Conditional Heteroskedasticity, In *Journal of Econometrics* vol. 31, 307-327.

[6] Breiman, Leo, 2001, Random Forests, In *Machine learning* 45, 5–32.

[7] Breiman, Leo, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone, 1984, Classification and regression trees, CRC press.

[8] Campbell, John Y. and Samuel B. Thompson, 2008, Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average?, In *The Review of Financial Studies* 21, No. 4, 1509-1531.

[9] Cederburg, Scott, Michael S. O'Doherty, Feifei Wang, Xuemin (Sterling) Yan, 2019, On the performance of volatility-managed portfolios, Forthcoming In *Journal of Financial Economics*.

[10] Engle, Robert, 1982, Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation In *Econometrica* vol. 50, 987-1008.

[11] Fama, Eugene and Kenneth R. French, 1988b, Dividend yields and expected stock returns, In *Journal of Financial Economics* 222, 3-25.

[12] Fleming, Jeff, Chris Kirby and Barbara Ostdiek, 2001, The Economic Value of Volatility Timing, In *The Journal of Finance* 56, 329-352.

[13] Frazzini, Andrea, Ronen Israel, and Tobias Moskowitz, 2015, Trading costs of asset pricing anomalies, Working paper, AQR Capital Management.

[14] Goyal, Amit and Ivo Welch, 2008, A Comprehensive Look at The Empirical Performance of Equity Premium Prediction, In *The Review of Financial Studies* 21 No. 4, 1455 - 1508.

[15] Gu, Shihao, Bryan T. Kelly, and Dacheng Xiu, 2018, Empirical Asset Pricing via Machine Learning, Chicago Booth Research Paper No. 18-04; 31st Australasian Finance and Banking Conference.

[16] Hallac, David, Peter Nystrup, and Stephen Boyd, 2018, Greedy Gaussian segmentation of multivariate time series, In *Advances in Data Analysis and Classification* 13 No. 3, 727–751.

[17] Hastie, Trevor, Jerome Friedman, and Robert Tibshirani, 2017, The Elements of Statistical Learning 2nd ed., Springer.

[18] Henriksson, Roy D, and Robert Merton, 1981, On Market Timing and Investment Performance. II. Statistical Procedures for Evaluating Forecasting Skills, In *The Journal of Business* 54 No. 4, 513-33.

[19] Jensen, Michael C., 1968, The Performance of Mutual Funds in the Period 1945-1964, In *The Journal of Finance* 23 No. 2, 389-416.

[20] Johannes, Michael, Arthur Korteweg, and Nicholas Polson, 2014, Sequential Learning, Predictability, and Optimal Portfolio Returns, In *The Journal of Finance* 69 No. 2, 611-644.

[21] Johannes, Michael, Nicholas Polson, and Jon Stroud, 2004, Sequential optimal portfolio performance: Market and volatility, Working Paper, Columbia University, University of Pennsylvania, and University of Chicago.

[22] Kandel, Shmuel and Robert F. Stambaugh, 1995, On the Predictability of Stock Returns: An Asset-Allocation Perspective NBER Working Paper #4997.

[23] Kirby, Chris and Barbara Ostdiek, 2012, It's All in the Timing: Simple Active Portfolio Strategies that Outperform Naive Diversification, In *Journal of Financial and Quantitative Analysis* 47 No. 2, 437–467.

[24] Liaw, Andy, and Matthew Wiener, 2002, Classification and Regression by random-Forest, In *R News* 2, 18-22.

[25] Liu, Fang, Xiaoxiao Tang, Guofu Zhou, 2019, Volatility-Managed Portfolio: Does It Really Work?, In *The Journal of Portfolio Management* 107.

[26] Markowitz, Harry, 1952, Portfolio Selection *The Journal of Finance*, Vol. 7, No. 1., 77-91.

[27] Matteson, David S. and David Ruppert, 2015, Statistics and Data Analysis for Financial Engineering, 2nd Ed .

[28] Merton, Robert, 1969, Lifetime Portfolio Selection under Uncertainty: The Continuous-Time Case In *The Review of Economics and Statistics* 51 No. 3, 247-57.

[29] Merton, Robert, 1981, On Market Timing and Investment Performance. I. An Equilibrium Theory of Value for Market Forecasts In *The Journal of Business* vol. 54, no. 3, 363-406.

[30] Moreira, Alan and Tyler Muir, 2017, Volatility-Managed Portfolios, In *The Journal of Finance* 69 No. 2, 1611-1644.

[31] Moreira, Alan and Tyler Muir, 2019, Should Long-Term Investors Time Volatility?, In *The Journal of Financial Economics* 131 No. 3, 507-527.

[32] Murphy, Kevin, 2012, Machine Learning - a Probabilistic Perspective, MIT Press.

[33] Samuelson, Paul A., 1969, Lifetime Portfolio Selection by Dynamic Stochastic Programming, In *Review of Economics and Statistics* vol. 51, 239-246.

[34] White, Halbert, 1980, A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity, In *Econometrica* vol. 48, 817-838.

[35] Yoo, Paul D., Maria H. Kim, and Tony Jan, 2005, Machine Learning Techniques and Use of Event Information for Stock Market Prediction: A Survey and Evaluation, International Conference on Computational Intelligence for Modeling, Control and Automation.

Appendix

A Additional Derivations

The base reward-risk time-series regression is given by

$$\frac{dR_t^{'}}{dt} - r_t^f = \alpha + \beta(\frac{dR_t}{dt} - r_t^f) + \epsilon_t, \tag{A.35}$$

with $dR_t^{'}$ given by Eq. 30. Next, define $f_t = \frac{dR_t}{dt} - r_t^f$ and $f_t^{'}$ as the left-hand side to get

$$f_t^{'} = \alpha + \beta f_t + \epsilon_t \tag{A.36}$$

To derive β, minimize the sum of squared residuals.

$$\min_{\alpha,\beta} E[(f_t^{'} - (\alpha + \beta f_t))^2]. \tag{A.37}$$

Solving the standard first-order conditions gives,

$$\beta = \frac{cov[f_t^{'}, f_t]}{var[f_t]} \tag{A.38}$$

For the base reward-risk timing,

$$\beta = \frac{cov[(r_t - r_t^f) \cdot \overline{r - r^f}/(\gamma\sigma_t^2), r_t - r_t^f]}{var[r_t - r_t^f]} = \frac{\overline{r - r^f}}{\gamma E[\sigma_t^2]} \tag{A.39}$$

For the machine learning reward-risk timing,

$$\beta = \frac{cov[(r_t - r_t^f) \cdot E[r_t - r_t^f|\mathcal{F}_{t-1}]/(\gamma E[\sigma_t^{*2}|\mathcal{F}_{t-1}]), r_t - r_t^f]}{var[r_t - r_t^f]}$$

$$= \frac{E[r_t - r_t^f|\mathcal{F}_{t-1}]}{\gamma E[\sigma_t^{*2}|\mathcal{F}_{t-1}]} \tag{A.40}$$

B Additional Tables

Table B1 shows the results for using only one machine learning model at a time in risk-adjusted returns. One variant is using a Random Forest model for the excess return and the realized one-month volatility for risk estimate. The other is substituting the expanding window mean for the excess return and a Random Forest to choose the optimal reference window length for the volatility. The additive benefit of machine learning risk premia forecasts over the expanding historical average are large. While the improvement in Sharpe ratios when volatility-timing with a varying window length instead of a constant one month is smaller, it is economically significant. The reason this benefit is smaller is because the range of values the volatility estimate can take is narrower, restricted by some average of recent realized volatilties. On one hand, this reduces the impact of any incorrect forecasts, but on the other, possibly leaves out potential alpha compared to directly forecasting the conditional volatility next month. Below are the exact weights for the two additional strategies.

- $w_7 = max(min(E_{RF}[r - r^f|\mathcal{F}_{t-1}]/(\gamma \cdot \sigma^2_{t-1}), 1.5), 0)$. This is using Random Forest for the reward estimate and the past month's realized variance for the risk, with a leverage limit of 50%.

- $w_8 = max(min(E[r - r^f]/(\gamma \cdot E_{RF}[\sigma^{*2}|\mathcal{F}_{t-1}]), 1.5), 0)$. This is using regression for the reward estimate and Random Forest for the risk estimate and a leverage limit of 50%.

The machine learning models in terms of Sharpe ratios are 'super-additive'. The sum of the improvements in Sharpe ratios relative to the base reward-risk timing when using the standalone models, 0.13, is smaller than the improvement when using them together for portfolio allocation, 0.14. This difference, however, is not statistically significant via the two-sample paired Sharpe ratios test (Bao, 2009).

Table B1: Additivity of Reward & Risk Machine Learning Models

In this table are the out-of-sample annual returns, standard deviations, and Sharpe ratios from 1952 to 2010 for the standalone Random Forest models. One variant is using a Random Forest model for the risk premia and the realized one-month volatility for risk estimate. The other is substituting the expanding window mean for the excess return and a Random Forest to choose the optimal reference window length for the volatility.

Sample	Strategy	Annual Return (%)	Standard Deviation (%)	Sharpe Ratio
1952 - 2010	Base Reward-Risk Timing (w_3)	11.57	14.82	0.46
	w_7	13.25	14.49	0.58
	w_8	12.03	15.57	0.47
	RF Reward-Risk Timing (w_1)	12.81	14.95	0.60

The ensemble of machine learning algorithms is slightly superior to each in separate, meaning there are non-linear interactions between the models themselves. Intuitively, when both models agree that the excess return will be positive, for instance, the weight is adjusted upwards according to the optimal weight, yet the higher combined market exposure than the exposure given by each model separately means the strategy is likely to capture the large positive returns, since the chances that both models are wrong is not greater than the chances one is. Additionally, the increase in the market portfolio weight is greater when using the models together than the sum of the increases by each respective model necessarily. When the models do not agree on the future excess return, they have opposing effects on the weight, yet the 'say' of the Random Forest model that gives the excess return estimate is measured by its confidence. Thee predictions of the two 'experts' are balanced out. Therefore, there may be benefits from using the reward and risk Random Forest models together, which is a question to explore in a future work.

C Decision Tree Algorithms

Algorithm C1 details how to build a classification tree and is a greedy algorithm (Breiman et al., 1984). We refer to the recursive version in (Murphy, 2012).

Algorithm C1: Classification Tree

Initialize stump node, $N_1(0)$. $N_k(d)$ is the kth node at depth d. S denotes the data, and C is the set of unique labels.

function fitTree($N_k(d)$, S, d)

1. The prediction of the $N_k(d)$ node is the majority vote of its observations, $sign(\sum_{i \in N_k(d)} y_i)$
2. Define the cost function as the Gini index. $cost(\{x_{ij}, y_{ij}\}) = \sum_{c=1}^{|C|} \hat{\pi}_c(1 - \hat{\pi}_c)$, where $\hat{\pi}_c$ is the frequency an entry in the leaf belongs to class c.
3. Select the optimal split: $(j^*, t^*) = \arg\min_{j \in \{1,..,|C|\}} \min_{t \in \mathcal{T}_j}(cost(\{x_{ij}, y_{ij} : x_{ij} \leq t\}) + cost(\{x_{ij}, y_{ij} : x_{ij} > t\}))$. $S_{left} = \{x_{ij}, y_{ij} : x_{ij} \leq t\}$, $S_{right} = \{x_{ij}, y_{ij} : x_{ij} > t\}$.
4. **if** *notworthSplitting(d, $cost$, S_{left},S_{right})* **then**
   return $N_k(d)$

   else

   Update the nodes:
   $N_1(d+1)$ = fitTree($N_k(d)$, S_{left}, $d+1$)
   $N_2(d+1)$ = fitTree($N_k(d)$, S_{right}, $d+1$)

   return $N_k(d)$

   end

Result: The classification tree model $f(\vec{x}) = \sum_{m=1}^{2^d} w_m \mathbb{1}\{\vec{x} \in S_m\}$, where $w_m = sign(\sum_{i \in S_m} y_i)$

The function notworthSplitting(d, $cost$, S_{left},S_{right}) contains stopping heuristics to prevent overfitting. In our case, the function value is true if the fraction of examples in either S_{left} or S_{right} is less than s_{min}, the minimum fraction of observations in a node determined by the user's parameter optimization.